(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 June 2002 (13.06.2002)

PCT

(10) International Publication Number
WO 02/47351 A2

(51) International Patent Classification[7]: H04L 29/00

(21) International Application Number: PCT/US01/47741

(22) International Filing Date:
10 December 2001 (10.12.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/733,587 8 December 2000 (08.12.2000) US

(71) Applicant: **TRUEDISK.COM, INC.** [US/US]; Suite 110, 7431 N.W. Evergreen Parkway, Hillsboro, OR 97124 (US).

(72) Inventors: **STEERE, David, C.**; 15700 S.W. Bridle Hills Drive, Beaverton, OR 97007 (US). **OTTO, Steven, W.**; 930 N.W. 12th, Apt. 528, Portland, OR 97209 (US). **MCNAMEE, Dylan, J.**; 143 S.E. 32nd Avenue, Portland, OR 97214 (US). **STUPAK, Michael**; 2328 N.W. Glisan #4, Portland, OR 97210 (US).

(74) Agents: **DIEHL, Robert, A.** et al.; Columbia IP Law Group, PC, Suite 820, 10260 S.W. Greenburg Road, Portland, OR 97223 (US).

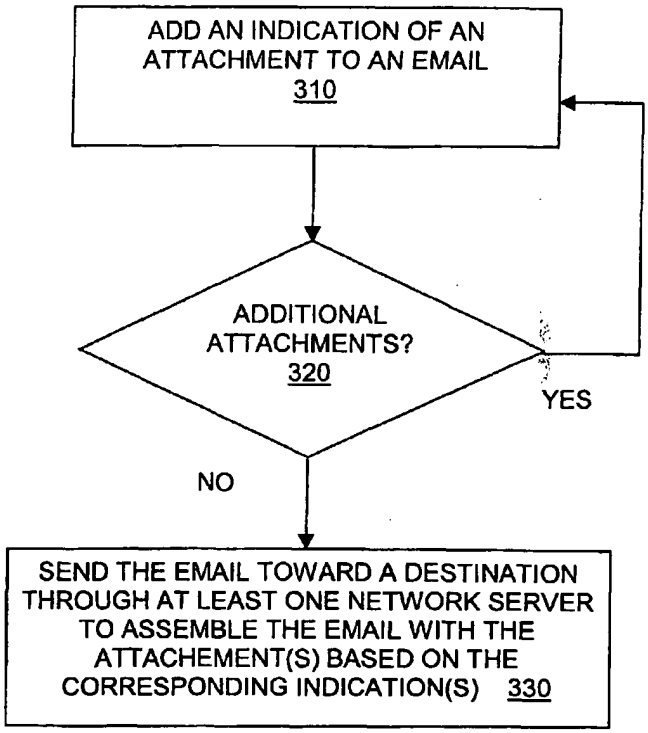
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW),

[Continued on next page]

(54) Title: MULTI-STAGE MESSAGE ASSEMBLY

(57) Abstract: Multistage electronic message assembly includes adding an indication of an attachment to an email at a network client. The network client then sends the email to a network server. The network server assembles the email with the attachment based at least in part on the indication.



WO 02/47351 A2

Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii)) for all designations*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii)) for all designations*

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

MULTI-STAGE MESSAGE ASSEMBLY

Field of the Invention

The present invention pertains to the field of electronic communications. More particularly, this invention relates to multi-stage assembly of electronic messages.

Background

Email and other forms of electronic messages have become part of everyday life for millions of people. Electronic messaging is often the most convenient way to communicate with others. Various forms of electronic messaging are used in desk top computers, lap top computers, programmable data assistants, cellular phones, pagers, and a variety of other devices.

Many electronic messaging formats allow users to send and receive more than just messages. Using these "rich" messaging formats, a user can often enhance electronic messages with attachments. Attachments may include soft copies of documents, pictures, executable software such as games or animation, and just about anything else that can be stored and transferred in electronic form.

When an attachment is added to an electronic message, the electronic message and the attachment are "assembled" at a host machine before the message is sent. For instance, a user may open a new email message on his or her computer, type a destination for the message, type a short note instructing the recipient to read an attached document, and then select a command to attach the document from an indicated location, such as the computer's local memory or a remote file server. The email application will then locate the indicated attachment, copy it, and combine the copied attachment with the rest of the email message. Once the message is assembled, it is ready to be sent. Any number of formats can be used to assemble messages, including the multi-purpose Internet extensions (MIME) format or extensible markup language (XML).

One potential drawback to attachments is speed. An attachment can be quit large. For instance, a few digital pictures can add up to several megabytes

of data. Assembling several megabytes of data into an email can take a comparatively long time, especially if the attachments are not stored in local memory. Once an email is assembled, sending an email that is several megabytes long can also take a comparatively long time.

Speed is of particular concern when slower network connections are used. For instance, if an employer provides a file server and an email server at a central office, an off-site employee may be able to use a modem to dial in to the email server over a telephone line. If the employee sends an email using the email server and includes an attachment that is located on the file server, the employee's machine will download a copy of the attachment from the file server at the central office over the telephone line, assemble the attachment into the email message, and then upload the assembled email message back to the email server at the central office.

Depending on the quality of the telephone connection, the employee's dial-up connection may typically transmit data in a range from about 10 to 56 thousand bits per second. In which case, even a relatively small attachment of approximately 100 thousand bytes could take from 14 to 80 seconds to down load from the file server. Larger attachments, like several megabytes worth of digital pictures, could take hours to down load. Of course, once an attachment has been downloaded and assembled, the assembled email will likely take just as long or longer to up load back to the email server.

The time delays caused by downloading remote attachments and uploading assembled messages are not as noticeable when faster network connections are used. However, even with the fastest network connections, at least some delay is introduced and potentially valuable network bandwidth is consumed.

BRIEF DESCRIPTION OF THE DRAWINGS

Examples of the present invention are illustrated in the accompanying drawings. The accompanying drawings, however, do not limit the scope of the present invention. Similar references in the drawings indicate similar elements.

Figure 1 illustrates one embodiment of the present invention.

Figure 2 illustrates another embodiment of the present invention.

Figure 3 demonstrates one embodiment of the present invention from the perspective of a client.

Figure 4 demonstrates one embodiment of the present invention from the perspective of a server.

Figure 5 illustrates yet another embodiment of the present invention including various features such as a server with a distributed and secure storage in an insecure network.

Figure 6 illustrates one embodiment of a hardware system to perform the present invention.

Figure 7 illustrates one embodiment of a machine readable storage medium to store machine readable instructions of the present invention.

DETAILED DESCRIPTION

In the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, those skilled in the art will understand that the present invention may be practiced without these specific details, that the present invention is not limited to the depicted embodiments, and that the present invention may be practiced in a variety of alternate embodiments. In other instances, well known methods, procedures, components, and circuits have not been described in detail.

Parts of the description will be presented using terminology commonly employed by those skilled in the art to convey the substance of their work to others skilled in the art. Also, parts of the description will be presented in terms of operations performed through the execution of programming instructions. As well understood by those skilled in the art, these operations often take the form of electrical, magnetic, or optical signals capable of being stored, transferred, combined, and otherwise manipulated through, for instance, electrical components.

Various operations will be described as multiple discrete steps performed in turn in a manner that is helpful in understanding the present invention. However, the order of description should not be construed as to imply that these operations are necessarily performed in the order they are presented, or even

order dependent. Lastly, repeated usage of the phrase "in one embodiment" does not necessarily refer to the same embodiment, although it may.

As used herein, "server" generally refers to one or more machines running server software to service requests received from other machines. Conversely, "client" generally refers to one or more machines running client software to request service from a server. Furthermore, "email" generally refers to any of a wide variety of electronic messaging formats including, but not limited to, internet email and instant network messaging. "Attachment" generally refers to any of a wide variety of email extensions.

Various embodiments of the present invention improve "rich" email by reducing time delays and network traffic caused by downloading attachments and uploading assembled email. In general, the present invention provides multi-stage email assembly. Attachments can be added to an email message en route, after the email has been sent. For instance, an email can be assembled with an attachment where access to the attachment is more convenient, efficient, and/or secure.

Figure 1 illustrates one embodiment of the present invention. Network client 110 sends email through email server 120 to destination 130. Network client 110 also stores files on file server 125. If network client 110 wants to email a file from file server 125 to destination 130, rather than downloading the file to network client 110 from the file server 125 and assembling the email with the attachment, network client 110 sends an email to email server 120 including an indication of, or pointer to, the file. When email server 120 receives the email, the email server can download the file based on the indication and assemble the email before forwarding it on to destination 130.

Assuming the network connection between email server 120 and file server 125 is faster than the network connection between client 110 and the two servers, assembling the email at email server 120 rather than at client 110 provides an overall performance improvement. That is, the time delay from when the client begins preparing the email to the time the email is received at the destination is reduced, providing increased efficiency. Downloading the file

from the file server takes less time, and, since the unassembled email is smaller, it takes less time to upload from the client to the email server.

Even if the network connection between email server 120 and file server 125 is slower than the network connection between client 110 and the two servers, assembling the email at the email server 120 rather than at client 110 provides a performance improvement from the perspective of the client. That is, the email may ultimately take longer to arrive at the destination if assembling the email at the email server takes longer than at the client. The client, however, does not have to wait for the file to download, and, since the unassembled email is smaller, it takes less of the client's time to upload to the email server, providing increased convenience for the client.

Multistage message assembly can also provide increased security. Email server 120 and file server 125 may be more secure than client 110. For instance, the servers may be part of a secure private local area network (LAN) and the client may be connected to the private LAN through the public, insecure Internet. Any number of sophisticated security approaches can be used to limit access to the private LAN. The client, however, may not be as secure. Downloading a copy of an attachment to the client may increase the likelihood of unauthorized access to the attachment. Using multistage message assembly however, the attachment can be stored and assembled within the private LAN where security is better.

Efficiency and convenience are of particular concern for clients having limited and/or expensive network connectivity or bandwidth. For instance, the rate at which data can be sent to and from a cellular phone over a wireless connection is likely to be very slow compared to the data rate among servers within a wire line private LAN. Moreover, wireless network connectivity to a cellular phone is likely to cost considerably more compared to a wire line private LAN. In which case, multistage email assembly can provide significant cost and convenience improvements for such clients .

Any number of approaches can be used to identify an indication of an attachment, locate the attachment, and assemble the email with the attachment. Any number of formats can be used for an indication of an

attachment and for assembled email. Various embodiments are described below.

Figure 2 illustrates another embodiment of the present invention in which an email is forwarded through multiple network servers prior to reaching its destination. Network client 210 uploads an email 215 with some number, M, indications of attachments. That is, client 210 intends for multiple attachments to be attached to the email. One or more of the attachments are remotely stored at one or more network servers. Rather than downloading all the attachments and assembling the email locally, client 210 adds an indication of each of the remotely stored attachments to email 215 and uploads the unassembled email to the first network server 220.

Network server 220 has associated with it a memory space 225. Any of a variety of memory configurations can be used for memory space 225. For instance, memory space 225 may be a hard drive located on the same machine that is running network server 220. Alternately, memory space 225 could be a separate file server running on a separate machine. As yet another example, memory space 225 could be a distributed file server in which memory resources are spread across multiple networked machines.

Based on the M indications, network server 220 can check memory space 225 to see if any of the corresponding attachments are available. If some number, X, of the attachments are available, server 220 can assemble the email with the attachments. In the illustrated embodiment, the server also updates the list of indications by removing indications for assembled attachments. This is particularly important where an attachment is available from multiple network servers where, for instance, two network servers along the path between client 210 and network destination 250 share some portion of a memory space. If the list of indications were not updated in this situation, multiple copies of an attachment may be added to the email. In alternate embodiments, any number of approaches can be used to prevent multiple copies of an attachment from being added to an email. For instance, a network server could check both the list of indications and the list of already assembled attachments.

The partially assembled email 227, now having M minus X indications, and X attachments, is forwarded to the second network server 230. As with the first network server, network server 230 checks its associated memory space 235 for attachments corresponding to the remaining indications. If one or more attachments are available, the email is assembled with the attachments, the indication list is updated, and the email is forwarded to the next network server.

The email can pass through any number, N, of network servers, one or more of which may attach one or more attachments. After passing through the N_{TH} network server 240, having memory space 245, the email 247 arrives at the network destination 250. Assuming all of the attachments were successfully located and assembled, email 247 arrives having zero indications and M assembled attachments.

In various embodiments, an indication of an attachment not only indicates a storage location for the attachment but also indicates a network server to assemble the attachment. In which case, client 210 may use a known network topology to plan a route for an email so that the email passes through each indicated network server. For instance, indications of attachments and network servers could be listed in a particular order. A network server could use the list not only to assemble available attachments, but also to route the email to the next listed server. Client 210 may perform various optimization functions to route an email through particular network servers that have the most efficient or convenient access to particular attachments.

In alternate embodiments, an indication of an attachment may merely indicate a storage location for the attachment somewhere within a network. In which case, the various network servers may use a known network topology to route the email so that the email passes through network servers having access to the indicated attachments. The network servers could also perform various optimization functions to route an email through particular network servers that have the most efficient or convenient access to particular attachments.

Figure 3 demonstrates one embodiment of the present invention from the perspective of a client that is sending an email. At 310, the network client adds an indication of an attachment to an email. As discussed above, the indication may include an indication of a storage location for the attachment as well as an indication of a network server to assemble the email with the attachment. Also as discussed above, various optimization functions can be used to plan a route for an email through particular servers in the network. For instance, the client may identify which network server from a number of available network servers can most efficiently, conveniently, or securely assemble the particular attachment. In which case, the client can indicate the location of the identified network server and define a route for the email through the identified network server.

At 320, if there are additional attachments for the email, the client goes back to add the additional indications for the attachments. Additional indications may be inserted into a planned route for the email or may simply be added to the end of a list. For instance, if an attachment is most conveniently, efficiently, or securely assembled at a particular network server, the client may insert the indication of the attachment in a list of indications so as to optimally route the email through that particular server.

At 330, if there are no additional attachments for the email, the client sends the email toward a destination. The email will be routed through at least one network server to assemble the email with the attachment(s) based on the corresponding indication(s). As discussed above, the email may be routed as determined by the client or as determined by the server(s).

Figure 4 demonstrates one embodiment of the present invention from the perspective of a network server. At 405, the server receives an email. If not for the present invention, the server would simply identify the next server along the path to the destination, or identify the destination itself if the destination were accessible from the server, and forward the email accordingly. The inventive server, however, does additional processing. At 410, if the email does not contain any indications of attachments, the email is forwarded toward its destination at 415 and the server returns to 405 to watch for the next email.

If the email does contain one or more indications, the server applies one of the indications to the memory space associated with the server at 420.

For instance, the server's memory space may include a database of namespaces and the indication may indicate a namespace from which the attachment can be downloaded. A namespace is commonly used in internet applications to create and share data using the Extensible Markup Language (XML). A namespace identifies a network address or Uniform Resource Locator (URL) at which information corresponding to the namespace can be found. In which case, applying the indication to the memory space could include searching the database of namespaces for an entry matching the namespace in the indication. In alternate embodiments, as discussed above, any number of different memory spaces can be used and, consequently, any number of approaches can be used to search a memory space for an attachment.

At 425, if an attachment is not located, the server proceeds to 445. At 445, if there is at least one additional indication, the server returns to 420 to process the next indication. If there are no additional indications, the email is forwarded toward its destination at 415 and the server returns to 405 to watch for the next email. If the attachment is located, the server identifies a source of the email and searches a security data structure for access authority to the attachment at 430. For instance, the server may use a return email address to identify the source of the email and compare that source to a list of sources authorized to access the attachment. In alternate embodiments, any number of approaches can be used to verify access authority.

At 435, if the source of the email does not have access authority to the attachment, the server proceeds to 445. At 445, if there is at least one additional indication, the server returns to 420 to process the next indication. If there are no additional indications, the email is forwarded toward its destination at 415 and the server returns to 405 to watch for the next email. If the source does have access authority, the sever retrieves the attachment, appends it to the email, and updates the list of indications at 440. As discussed above, any number of approaches can be used for each of these functions.

At 445, if there is at least one additional indication, the server returns to 420 to process the next indication. If there are no additional indications, the email is forwarded toward its destination at 415 and the server returns to 405 to watch for the next email.

Various alternate embodiments my include additional elements, may not include all of the illustrated element, and/or may not perform elements in the same order. For instance, in one embodiment, the server may send some form of error message to the source and/or destination of an email when and if the server determines that the source does not have access authority to a particular attachment. In another embodiment, both the source and destination of the email are checked for access authority before assembling the email with a given attachment.

In an embodiment in which the email is routed through multiple servers, the route may be optimized by either the client or the servers. In which case, a server may identify a next server in the route based, for instance, on the listed indications, and forward the email toward the next server. For example, the list of indications may specify which server should be next. Alternately, a server may choose which server should be next based purely on storage locations for the one or more indicated attachments or based on additional criteria, such as efficiency, convenience, and/or security.

Figure 5 illustrates one embodiment of the present invention involving a distributed network server in an insecure network 540. Network 540 includes a network server manager 510, a number of remote storage elements 520, and a number of network clients 530. In the illustrated embodiment, network server manager 510 and remote storage elements 520 together comprise a distributed network server to service requests from network clients 530. For instance, the distributed network server comprises a file server to store files for network clients 530. The files are stored in the remote storage elements 520 and access to the files is routed through network server manager 510.

Network server manager 510 provides a firewall 550, a database of namespaces 560 defining the memory space distributed among the remote storage elements 520, account management 570 to manage access to the

distributed memory space, and email server 515. Each network client 530 includes an email function 535 to send and receive email through the email server 515 of network server manager 510. Network server manager 510 assembles the emails with any indicated attachments that can be retrieved from the remote storage elements 520.

Since remote storage elements 520 are in an insecure network, the storage elements would normally be vulnerable to unauthorized access. Unauthorized access is typically gained by repeatedly sending various forms of requests for service to a network element until a request is found to which the network element will respond. In the illustrated embodiment however, the remote storage elements do not respond to any requests for service. Rather, the storage elements operate like clients in that they only issue requests and receive responses to their own requests, essentially rendering them immune to unauthorized access.

In order to allow network server manager 510 to retrieve attachments from the remote storage elements, each remote storage element includes a polling function 190. The polling function issues requests to the network server manager 510 at various intervals. Network server manager 510 includes a queuing function. Queues 580 are used to store requests from network clients, such as requests to retrieve attachments. When a request is received from a remote storage element, the network server manager 510 responds to the request and includes any network client requests that have been queued for that particular remote storage element, such as a request to retrieve an email attachment. When the remote storage element sends another request, the remote storage element will include responses to network client requests previously received.

Figure 6 illustrates one embodiment of a hardware system intended to represent a broad category of computer systems such as personal computers, workstations, and/or embedded systems. In the illustrated embodiment, the hardware system includes processor 610 coupled to high speed bus 605, which is coupled to input/output (I/O) bus 615 through bus bridge 630. Temporary memory 620 is coupled to bus 605. Permanent memory 640 is

coupled to bus 615. I/O device(s) 650 is also coupled to bus 615. I/O device(s) 650 may include a display device, a keyboard, one or more external network interfaces, etc.

Certain embodiments may include additional components, may not require all of the above components, or may combine one or more components. For instance, temporary memory 620 may be on-chip with processor 610. Alternately, permanent memory 640 may be eliminated and temporary memory 620 may be replaced with an electrically erasable programmable read only memory (EEPROM), wherein software routines are executed in place from the EEPROM. Some implementations may employ a single bus, to which all of the components are coupled, or one or more additional buses and bus bridges to which various additional components can be coupled. Those skilled in the art will be familiar with a variety of alternate internal networks including, for instance, an internal network based on a high speed system bus with a memory controller hub and an I/O controller hub. Additional components may include additional processors, a CD ROM drive, additional memories, and other peripheral components known in the art.

One component in particular that may be included in the hardware system of Figure 6 for implementation of the present invention is a random number generator for security purposes. For instance, communications among the network server manager 510 and the various remote storage elements 520 in Figure 5 can be encrypted to provide secure communications among the distributed elements. Security is only as strong as the keys used to encrypt the data. Keys are often generated using a deterministic algorithm based on some starting number. If the starting number is a predictable value, such as the current date or time of day, the cryptographic keys are more easily compromised, leaving the distributed storage vulnerable to attack. If the starting number is truly random and unpredictable, as provided by a random number generator, the communications are much more secure.

Continuing on with Figure 6, in one embodiment, the various elements of the present invention described above are each implemented using one or more computers such as the hardware system of Figure 6. Where more than

one computer is used, the systems can be coupled to communicate over an external network, such as a local area network (LAN), an internet protocol (IP) network, etc. In one embodiment, the present invention is implemented as software routines executed by one or more execution units within the computer(s). For a given computer, the software routines can be stored on a storage device, such as permanent memory 640.

Alternately, as shown in Figure 7, the software routines can be machine executable instructions 710 stored using any machine readable storage medium 720, such as a diskette, CD-ROM, magnetic tape, digital video or versatile disk (DVD), laser disk, ROM, Flash memory, etc. The series of instructions need not be stored locally, and could be received from a remote storage device, such as a server on a network, a CD ROM device, a floppy disk, etc., through, for instance, I/O device(s) 650 of Figure 6.

From whatever source, the instructions may be copied from the storage device into temporary memory 620 and then accessed and executed by processor 610. In one implementation, these software routines are written in the C programming language. It is to be appreciated, however, that these routines may be implemented in any of a wide variety of programming languages.

In alternate embodiments, the present invention is implemented in discrete hardware or firmware. For example, one or more application specific integrated circuits (ASICs) could be programmed with one or more of the above described functions of the present invention. In another example, one or more functions of the present invention could be implemented in one or more ASICs on additional circuit boards and the circuit boards could be inserted into the computer(s) described above. In another example, field programmable gate arrays (FPGAs) or static programmable gate arrays (SPGA) could be used to implement one or more functions of the present invention. In yet another example, a combination of hardware and software could be used to implement one or more functions of the present invention.

Thus, multi-stage electronic message assembly is described. Whereas many alterations and modifications of the present invention will be

comprehended by a person skilled in the art after having read the foregoing description, it is to be understood that the particular embodiments shown and described by way of illustration are in no way intended to be considered limiting. Therefore, references to details of particular embodiments are not intended to limit the scope of the claims.

CLAIMS

What is claimed is:

1. A method comprising:

adding an indication of an attachment to an electronic delivery unit at a network client; and

sending the electronic delivery unit to a network server, said network server to assemble the electronic delivery unit with the attachment based at least in part on the indication.

2. The method of claim 1 wherein the network server is further to route the electronic delivery unit with the attachment toward a destination.

3. The method of claim 1 wherein the electronic delivery unit comprises one of an email message and an instant network message.

4. The method of claim 1 wherein the network client comprises one of a desk top computer, a lap top computer, a personal data assistant, a set top box, a cellular phone, and a pager.

5. The method of claim 1 further comprising:

determining that the network server has access to the attachment;

identifying a location of the network server among a plurality of network servers in a network; and

defining the network server to assemble the electronic delivery unit based on the location.

6. The method of claim 5 wherein defining the network server to assemble the electronic delivery unit comprises at least one of:

including the location of the network server in the indication of the attachment; and

defining a route for the electronic delivery unit through the network including the network server.

7. The method of claim 5 wherein determining that the network server has access comprises:

identifying multiple network servers from among the plurality of network servers that have access to the attachment; and

selecting the network server to assemble the electronic delivery from among the multiple network servers based on a selection criteria.

8. The method of claim 7 wherein the selection criteria comprises at least one of efficient access to the attachment, convenient access to the attachment, and heightened security for the attachment.

9. A method comprising:

receiving an electronic delivery unit at a network server from a network client, said electronic delivery unit including an indication of an attachment for the electronic delivery unit; and

assembling the electronic delivery unit with the attachment based at least in part on the indication.

10. The method of claim 9 wherein assembling the electronic delivery unit comprises:

detecting the indication within the electronic delivery unit;

determining whether the attachment corresponding to the indication is available; and

retrieving the attachment and appending the attachment to the electronic delivery unit if the attachment is available.

11. The method of claim 10 wherein determining whether the attachment is available comprises:

applying the indication of the attachment to a memory space of the network server.

12. The method of claim 11 wherein the memory space comprises at least one of a local memory, a remote memory, and a distributed memory.

13. The method of claim 11 wherein the memory space is defined by at least one of an address space, a virtual address space, and a namespace.

14. The method of claim 11 wherein applying the indication of the attachment comprises:

searching the memory space of the network server for a memory location defined by the indication of the attachment.

15. The method of claim 10 wherein determining whether the attachment is available comprises:

determining if a source of the electronic delivery unit has authority to access the attachment.

16. The method of claim 15 wherein determining if the source of the electronic delivery has authority comprises:

identifying the source of the electronic delivery unit; and

applying the source to a security data structure that defines access authority for data available to the network server.

17. The method of claim 9 wherein a format of the attachment comprises one of a multi-purpose Internet extensions (MIME) format or an extensible markup language (XML).

18. The method of claim 9 further comprising:

routing the electronic delivery unit toward a destination.

19. The method of claim 18 wherein routing the electronic delivery unit comprises:

identifying a next hop for the electronic delivery unit based on a next indication.

20. The method of claim 18 wherein the indication of the attachment comprises a current indication, the method further comprising:

deleting the current indication from a list of indications prior to routing the electronic delivery unit.

21. The method of claim 18 wherein routing the electronic delivery unit comprises:

selecting a next hop for the electronic delivery unit based on a selection criteria.

22. The method of claim 21 wherein the selection criteria comprises at least one of efficient access to the attachment, convenient access to the attachment, and heightened security for the attachment.

23. A method comprising:

adding a plurality of indications of a plurality of attachments to an electronic delivery unit at a network client; and

sending the electronic delivery unit toward a destination through at least one network server, said at least one network server to assemble the electronic delivery unit with selected ones of the plurality of attachments based at least in part on corresponding ones of the plurality of indications.

24. A machine readable storage medium having stored thereon machine executable instructions to implement a method comprising:

adding an indication of an attachment to an electronic delivery unit at a network client; and

sending the electronic delivery unit to a network server, said network server to assemble the electronic delivery unit with the attachment based at least in part on the indication.

25. A machine readable storage medium having stored thereon machine executable instructions to implement a method comprising:

receiving an electronic delivery unit at a network server from a network client, said electronic delivery unit including an indication of an attachment for the electronic delivery unit; and

assembling the electronic delivery unit with the attachment based at least in part on the indication.

26. A client apparatus comprising:

a processor; and

a storage medium having stored there on executable instructions to be executed by the processor to add an indication of an attachment to an electronic delivery unit, and send the electronic delivery unit to a network server, said network server to assemble the electronic delivery unit with the attachment based at least in part on the indication.

27. A server apparatus comprising:

a processor; and

a storage medium having stored there on executable instructions to be executed by the processor to receive an electronic delivery unit from a network client, said electronic delivery unit including an indication of an attachment for the electronic delivery unit, and assemble the electronic delivery unit with the attachment based at least in part on the indication.

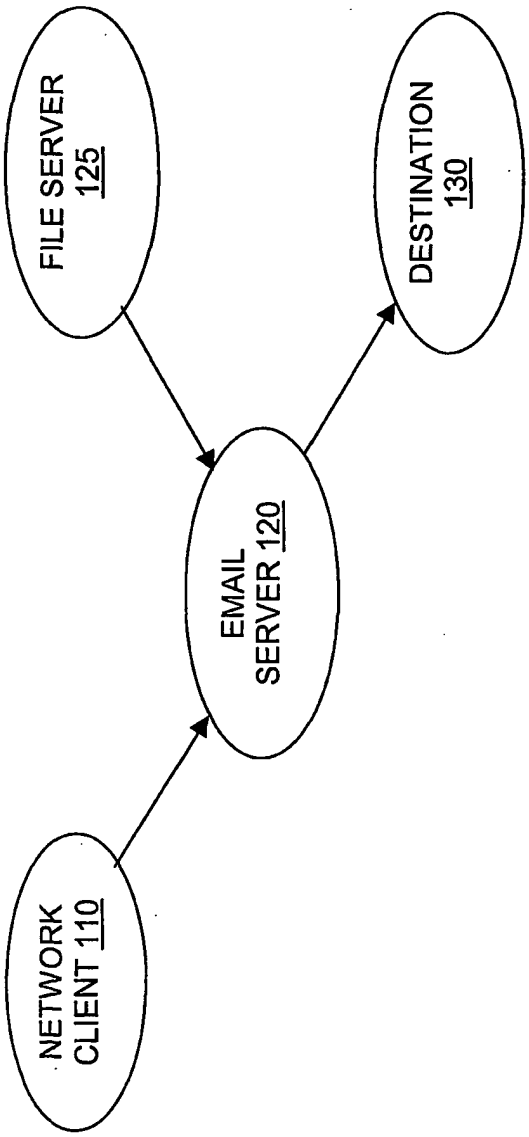

FILE SERVER 125
DESTINATION 130
EMAIL SERVER 120
NETWORK CLIENT 110

FIG.1

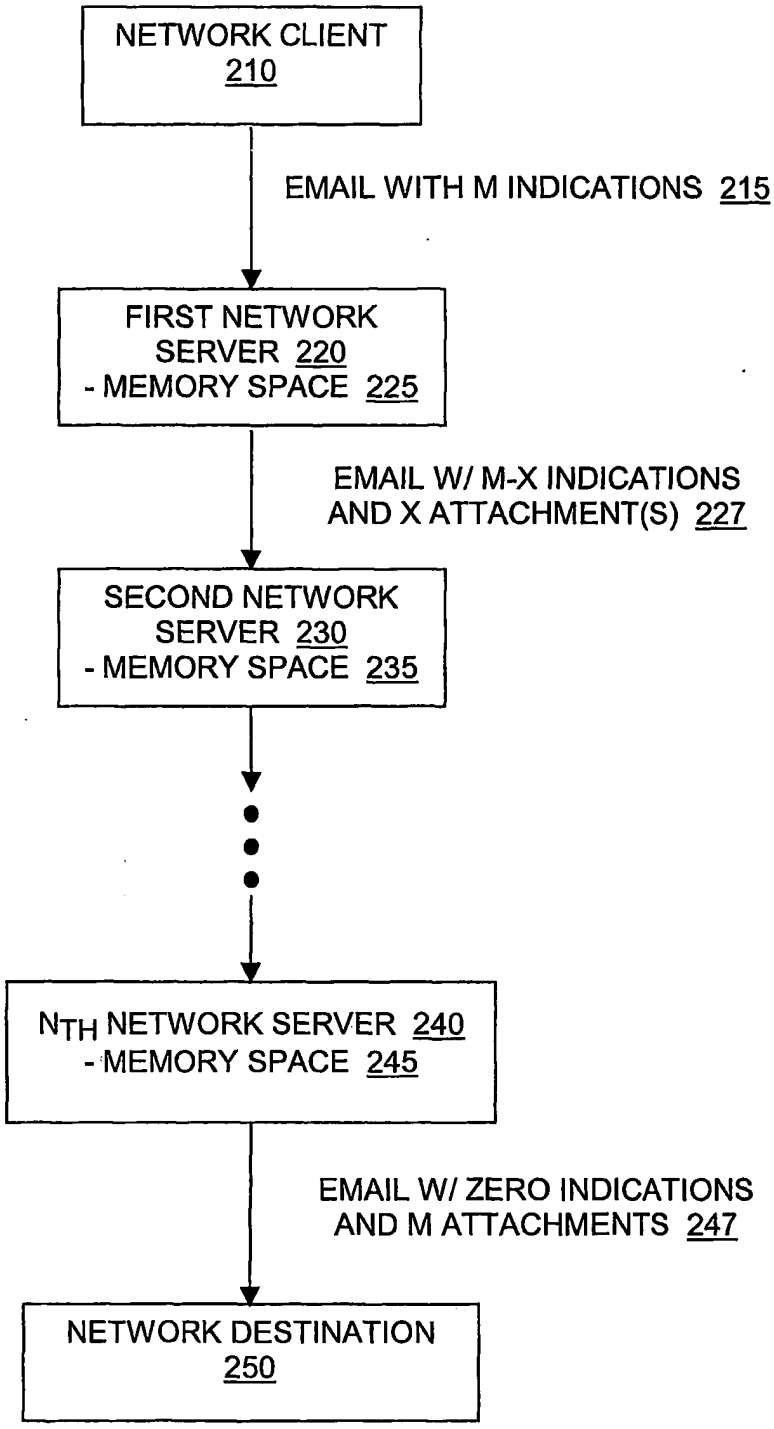

NETWORK CLIENT
210
EMAIL WITH M INDICATIONS 215
FIRST NETWORK
SERVER 220
- MEMORY SPACE 225
EMAIL W/ M-X INDICATIONS
AND X ATTACHMENT(S) 227
SECOND NETWORK
SERVER 230
- MEMORY SPACE 235
NTH NETWORK SERVER 240
- MEMORY SPACE 245
EMAIL W/ ZERO INDICATIONS
AND M ATTACHMENTS 247
NETWORK DESTINATION
250

FIG. 2

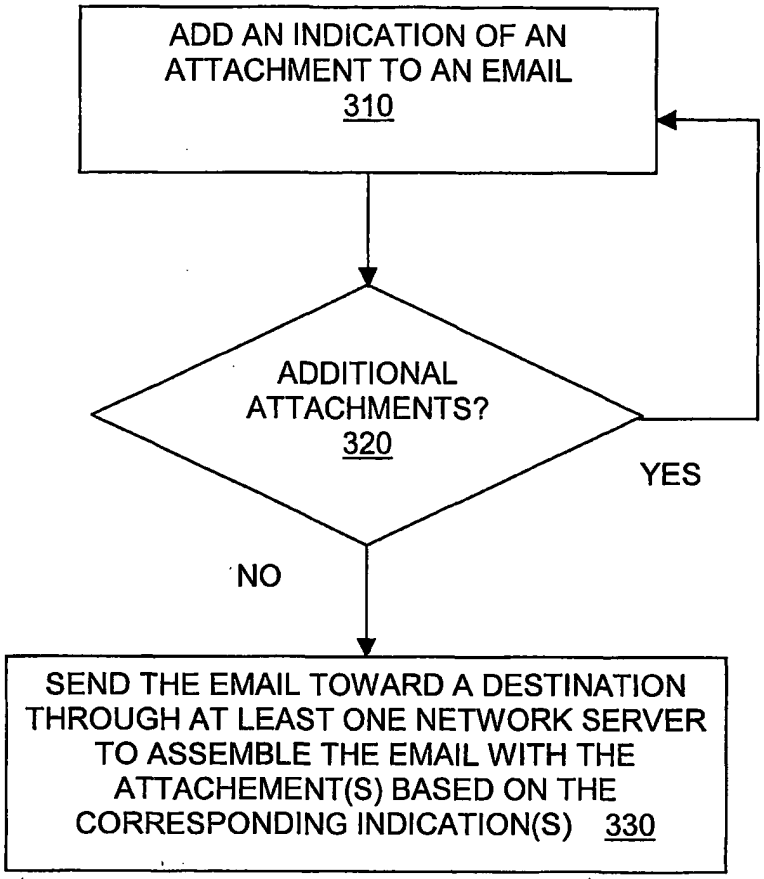

ADD AN INDICATION OF AN ATTACHMENT TO AN EMAIL
310
ADDITIONAL ATTACHMENTS?
320
YES
NO
SEND THE EMAIL TOWARD A DESTINATION THROUGH AT LEAST ONE NETWORK SERVER TO ASSEMBLE THE EMAIL WITH THE ATTACHEMENT(S) BASED ON THE CORRESPONDING INDICATION(S) 330

FIG. 3

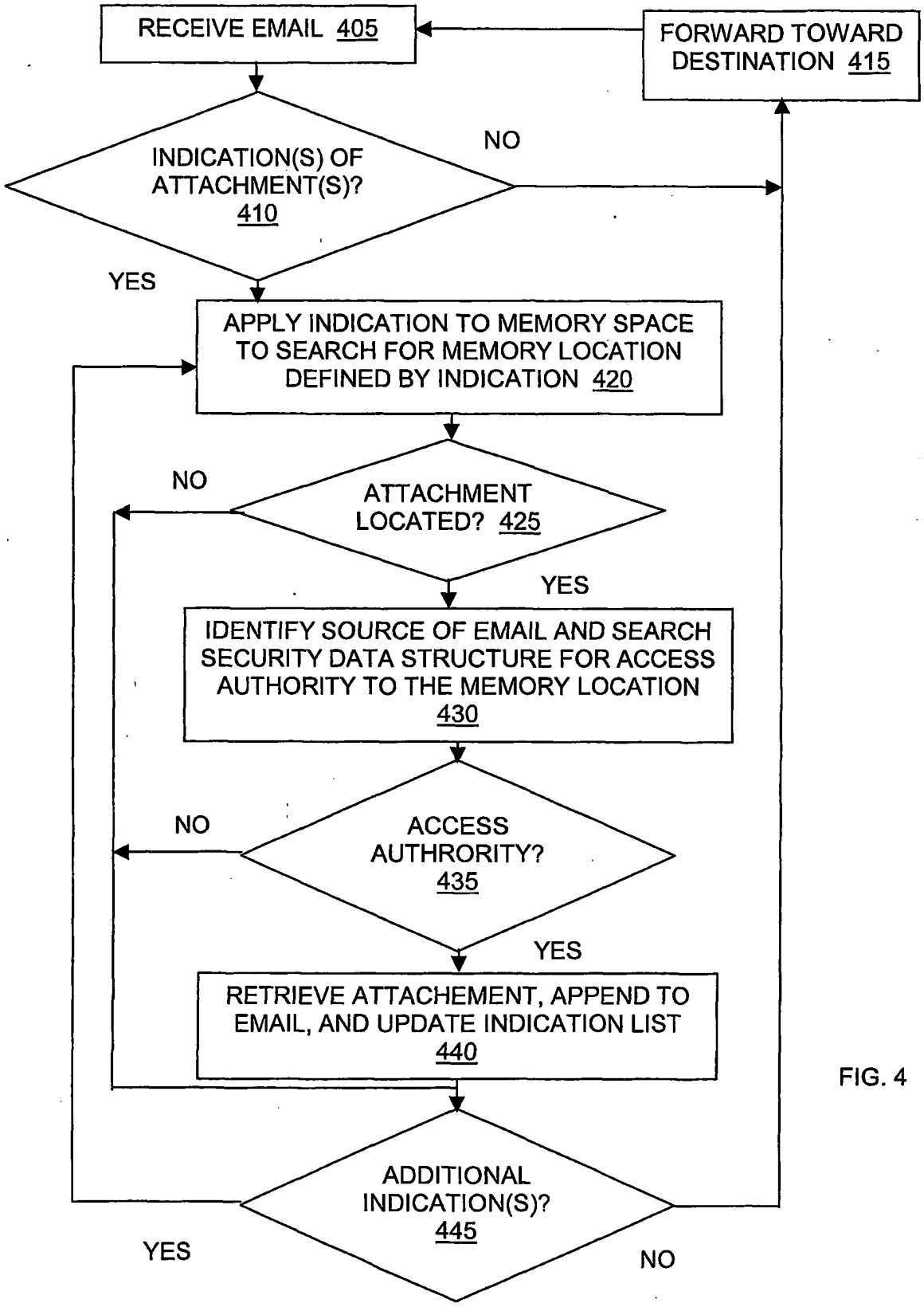

RECEIVE EMAIL 405
FORWARD TOWARD DESTINATION 415
INDICATION(S) OF ATTACHMENT(S)? 410
NO
YES
APPLY INDICATION TO MEMORY SPACE TO SEARCH FOR MEMORY LOCATION DEFINED BY INDICATION 420
ATTACHMENT LOCATED? 425
NO
YES
IDENTIFY SOURCE OF EMAIL AND SEARCH SECURITY DATA STRUCTURE FOR ACCESS AUTHORITY TO THE MEMORY LOCATION 430
ACCESS AUTHRORITY? 435
NO
YES
RETRIEVE ATTACHEMENT, APPEND TO EMAIL, AND UPDATE INDICATION LIST 440
ADDITIONAL INDICATION(S)? 445
YES
NO

FIG. 4

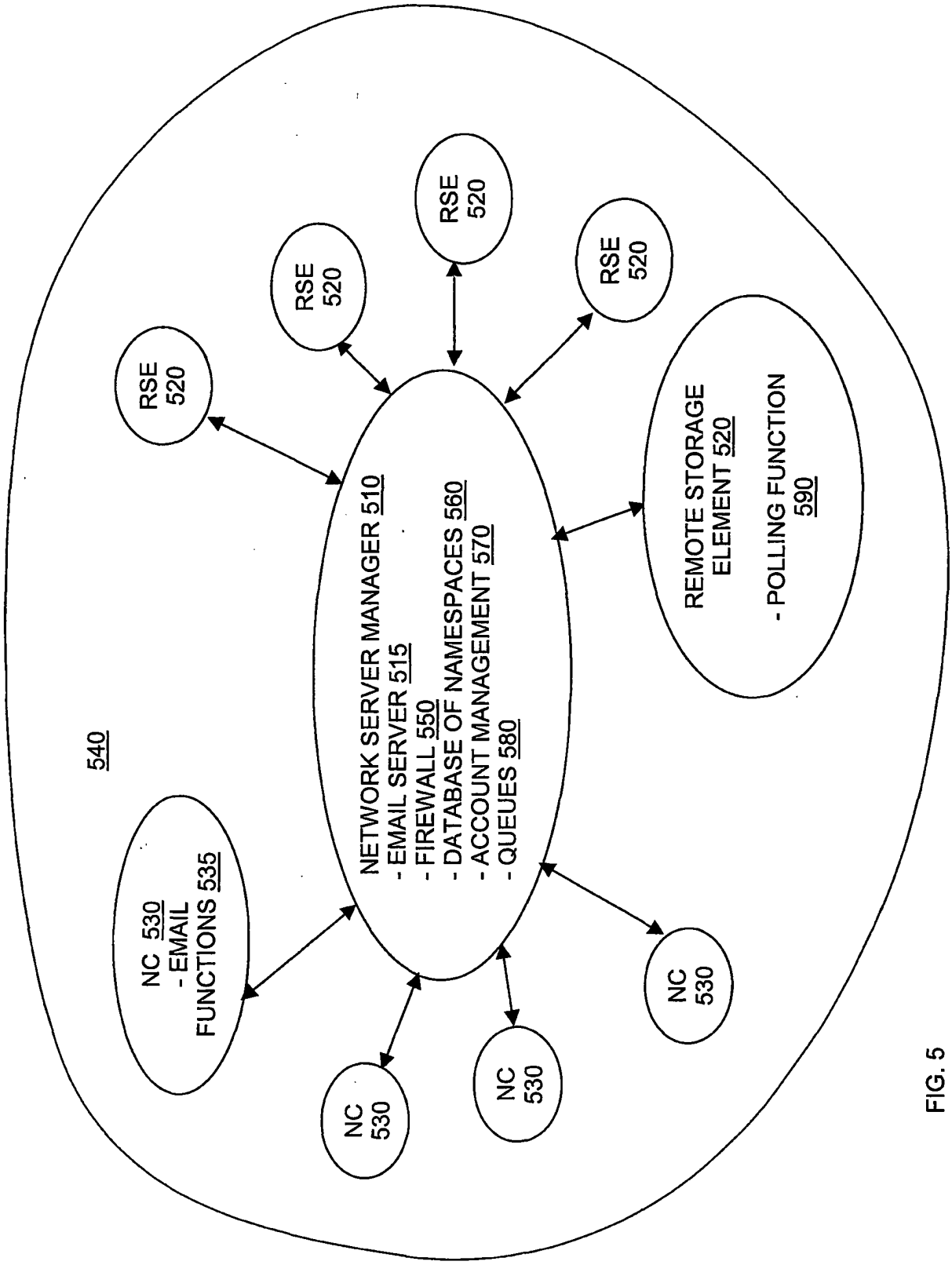

540
NC 530
- EMAIL
FUNCTIONS 535
NETWORK SERVER MANAGER 510
- EMAIL SERVER 515
- FIREWALL 550
- DATABASE OF NAMESPACES 560
- ACCOUNT MANAGEMENT 570
- QUEUES 580
REMOTE STORAGE
ELEMENT 520
- POLLING FUNCTION
590
RSE 520
RSE 520
RSE 520
RSE 520
NC 530
NC 530
NC 530

FIG. 5

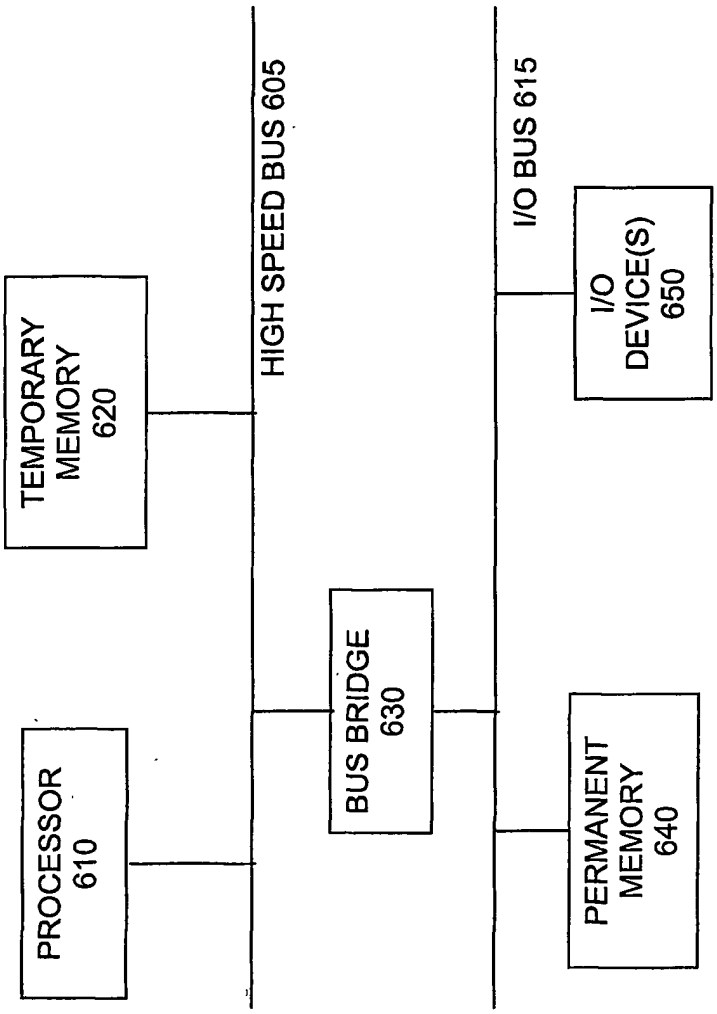

HIGH SPEED BUS 605
I/O BUS 615
TEMPORARY MEMORY 620
I/O DEVICE(S) 650
PROCESSOR 610
BUS BRIDGE 630
PERMANENT MEMORY 640

FIG. 6

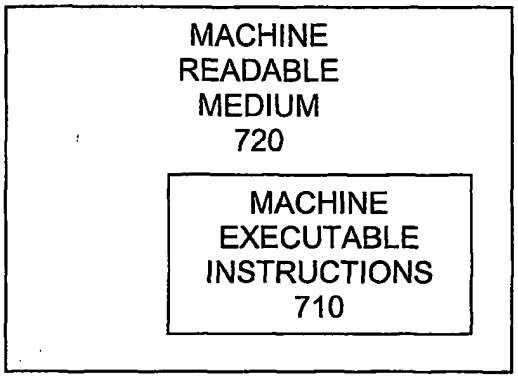

MACHINE
READABLE
MEDIUM
720
MACHINE
EXECUTABLE
INSTRUCTIONS
710

FIG. 7